Notice d'utilisation de la manipulation de champs

Résumé:

L'objectif de ce document est de fournir à l'utilisateur quelques recettes liées à la manipulation des champs.

Dans ce document sont décrits les cas suivants :

- Poursuivre un calcul `STAT_NON_LINE` après avoir retiré ou ajouté des éléments dans le modèle (ex : creusement d'un tunnel, mise en place des couches d'un barrage) ;
- Définir un état de contraintes initiales (ou de variables internes) à partir de « formules analytiques » ;
- Appliquer une pression sur une partie (non prévue dans le maillage) du bord d'une structure 3D ;
- Calculer l'intégrale des contraintes sur le bord d'une structure 3D.

Table des matières

1 Comment poursuivre un calcul STAT_NON_LINE après avoir retiré ou ajouté des éléments dans le modèle

1.1 Problématique

La reprise d'un calcul avec un modèle composé d'éléments supplémentaires ou au contraire d'un modèle restreint à certains éléments nécessite certaines précautions. En effet, le calcul n'est plus réalisé sur le même nombre d'éléments, ce qui signifie que les champs définissant l'état initial de reprise de calcul (champs de déplacement, de contraintes et de variables internes) doivent subir quelques ajustements pour pouvoir être pris en compte dans l'opérateur de résolution.

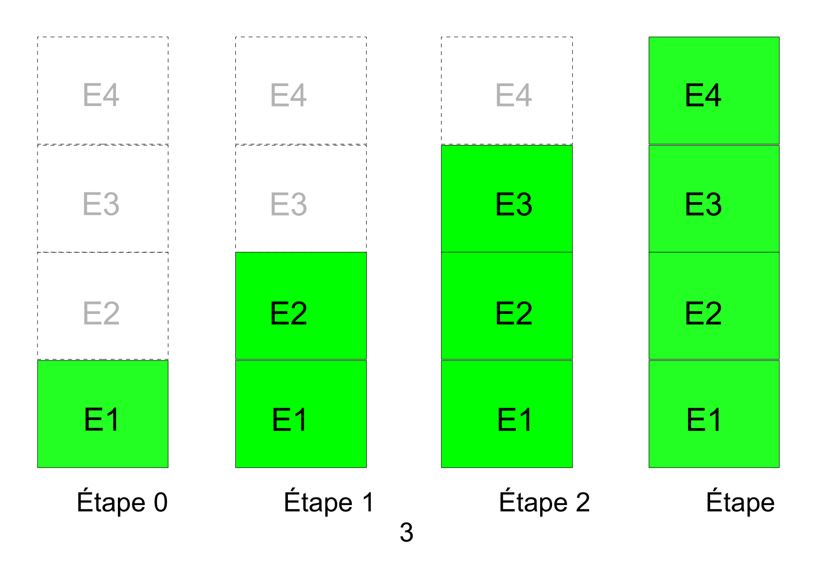
Quand on retire des éléments finis du modèle, le problème est a priori assez simple : il suffit d' « oublier » de l'information. En revanche, quand on veut ajouter des éléments, il faut en « inventer ».

C'est pour cela que dans la suite de ce document, nous traiterons le cas de l'ajout d'éléments.

1.2 Ajout d'éléments dans le modèle

1.2.1 Objectif

Nous allons présenter une méthodologie qui permet de réaliser ce genre de calcul à partir d'un exemple.



Dans cet exemple on souhaite simuler l'ajout progressif d'éléments dans l'étude comme par exemple simuler la construction d'un barrage en ajoutant successivement des couches d'éléments.

Quatre étapes seront nécessaires pour réaliser ces calculs : A la première étape, le calcul sera effectué sur un modèle composé du groupe de mailles $E1$, à la seconde sur un modèle composé des groupes de mailles $E1+E2$, et ainsi de suite jusqu'à l'étape 4. L'utilisateur aura pris soin lors de la construction du maillage de définir explicitement les groupes de mailles $E1$, $E2$, $E3$ et $E4$.

Il commencera par réaliser l'étape 0, qui consiste à effectuer le calcul sur le maillage restreint au groupe $E1$. Pour les étapes suivantes, qui consistent à poursuivre le calcul effectué précédemment, plusieurs « formes » de mise en œuvre sont envisageables :

- soit on effectue une poursuite (nouveau fichier de commande) : ce choix n'est pas pertinent si le nombre de couches d'éléments est conséquent,

- soit on enrichit le fichier de commande : la taille du fichier de commande est fonction du nombre de poursuites, et donc du nombre de couches; ce choix n'est pas forcément judicieux.
- soit on utilise une boucle python : nous privilégions ce choix car il a l'avantage d'être facilement évolutif pour prendre en compte l'ajout de couches supplémentaires.

Dans le « fond », la procédure est la même : dans le but de fournir un état initial au calcul, elle fait appel à l'opérateur `CREA_CHAMP` :

- pour extraire les champs de déplacements, de contraintes, et de variables internes,
- pour étendre ces champs au nouveau modèle.

Le schéma suivant illustre la structure simplifiée du fichier de commande:

```
⇒ Initialisation des listes python pour stocker les concepts Aster

⇒ Etape 0 :
   - création du GROUP_MA 'ELEM': ELEM = E1
   - création du modèle MOD[0], du champ matériau MAT[0], du
     chargement CHA[0].
   - réalisation du calcul U[0]
   - extraction des champs de U[0], utiles à l'état initial du calcul
     suivant
┌─ Boucle python sur i (i=1,3)pour traiter les étapes 1, 2 et 3 :
│
│   ⇒ Mise à jour du GROUP_MA 'ELEM': ELEM = E1+…+E(i+1)
│
│   ⇒ Création du modèle MOD[i], du champ materiau MAT[i] et du
│     chargement CHA[i].
│
│   ⇒ Création des champs sur le nouveau modèle à partir des
│     champs extraits à l'étape i-1
│
│   ⇒ Réalisation du calcul U[i] en ayant spécifié un état initial
│
└─  ⇒ Extraction des champs de U[i] pour l'étape i+1
```

Dans le tableau ci-dessous nous présentons les commandes exécutées ainsi que les concepts produits. On remarquera comment transitent les concepts d'une étape à l'autre.

		Étape 0	Étape 1	Étape 2	Étape 3
Maillage	LIRE_MAILLAGE DEFI_GROUP	E1	E1+E2	E1+E2+E3	E1+E2+E3+E4
Modèle	AFFE_MODELE	MOD_0	MOD_1	MOD_2	MOD_3
Matériau	AFFE_MATERIAU	MAT_0	MAT_1	MAT_2	MAT_3
Charge	AFFE_CHAR_MECA	CHA_0	CHA_1	CHA_2	CHA_3
État initial	CREA_CHAMP/ASSE		DEP_0* SIG_0* VARI_0*	DEP_1* SIG_1* VARI_1*	DEP_2* SIG_2* VARI_2*
Solution	STAT_NON_LINE	U_0	U_1	U_2	U_3
Extraction	CREA_CHAMP/EXTR	DEP_0 SIG_0 VARI_0	DEP_1 SIG_1 VARI_1	DEP_2 SIG_2 VARI_2	DEP_3 SIG_3 VARI_3

Tableau 1.1 : concepts produits

(*: champ calculé à l'étape i, attendu à l'étape i+1)

On présente dans la suite les différentes étapes et leurs mises en œuvre dans le fichier de commande.

1.2.2 Mise en œuvre

1.2.2.1 Initialisation

Chaque concept produit par ces opérateurs sera stocké dans une liste python. Il convient donc d'allouer ces listes. Soit n le nombre de couches d'éléments : Dans cet exemple, n=4.
On écrira dans le fichier de commande:

- `MOD=[None]*n,`
- `MAT=[None]*n,`
- `CHA=[None]*n,`
- `U=[None]*n,`
- `DEP=[None]*n,`
- `SIG=[None]*n,`
- `VARI=[None]*n.`

Remarque :
Il suffit de modifier la valeur de n pour prendre en compte de nouvelles couches d'éléments.

1.2.2.2 Étape 0: couche E1

- **on définit le groupe `'ELEM'` qui évoluera en fonction des différentes étapes :**

```
DEFI_GROUP(MAILLAGE=MA, CREA_GROUP_MA=_F(NOM='ELEM',GROUP_MA='E1'),)
```

- **Création du nouveau modèle, affectation du matériau et du chargement :**

```
MOD[0]=AFFE_MODELE(MAILLAGE=MA,AFFE=_F(GROUP_MA='ELEM',...)

MAT[0]=AFFE_MATERIAU(MODELE=MOD[0], AFFE=_F(TOUT='OUI',MATE=...),...)

CHA[0]=AFFE_CHAR_MECA(MODELE=MOD[0], …)
```

- **`Calcul :`**

```
U[0]=STAT_NON_LINE(MODELE=MOD[0],CHAM_MATER=MAT[0],
                   EXCIT=_F(CHARGE=CHA[0]),...)
```

- **Extraction des champs de déplacements, contraintes, variables internes :**

```
DEP[0]=CREA_CHAMP(TYPE_CHAM='NOEU_DEPL_R',
                 OPERATION='EXTR',
                 RESULTAT=U[0],
                 NOM_CHAM='DEPL',
                 ...);

SIG[0]=CREA_CHAMP(TYPE_CHAM='ELGA_SIEF_R',
                  OPERATION='EXTR',
                  RESULTAT=U[0],
                  NOM_CHAM='SIEF_ELGA',
                  ...);

VAR[0]=CREA_CHAMP(TYPE_CHAM='ELGA_VARI_R',
                  OPERATION='EXTR',
                  RESULTAT=U[0],
                  NOM_CHAM='VARI_ELGA',
                  ...);
```

1.2.2.3 Étape i : couches E1+...+Ei

A ce stade, les calculs ont déjà été réalisés aux étapes `0` à `i-1`. Nous nous trouvons dans la boucle python « `for i in range(1,n):` » à l'indice `i` :

- **Actualisation du groupe de mailles `'ELEM'` du maillage `MA`**

```
DEFI_GROUP(MAILLAGE=MA,
           CREA_GROUP_MA=_F(NOM='ELEM0',GROUP_MA='ELEM'),...)

DEFI_GROUP(MAILLAGE=MA, DETR_GROUP_MA=_F(NOM='ELEM'),...)

DEFI_GROUP(MAILLAGE=MA,
           CREA_GROUP_MA=(_F(NOM='ELEM', OPTION='UNION',
                             GROUP_MA=('ELEM0','E'+str(i+1)),),...)
```

- **Création du nouveau modèle, affectation du matériau et du chargement**

```
MOD[i]=AFFE_MODELE(MAILLAGE=MA,AFFE=_F(GROUP_MA='ELEM',...)

MAT[i]=AFFE_MATERIAU(MODELE=MOD[i], AFFE=_F(TOUT='OUI',MATE=...),...)

CHA[i]=AFFE_CHAR_MECA(MODELE=MOD[i], ...)
```

- **Création des champs de déplacements, contraintes et variables internes pour l'état initial de l'étape i**

```
DEPBID=CREA_CHAMP(TYPE_CHAM='NOEU_DEPL_R',
                      OPERATION='AFFE',
                      MODELE=MOD[i],
                      AFFE=_F(GROUP_MA='ELEM',
                              NOM_CMP=('DX','DY','DZ',...),
                              VALE=(1.0,1.0,1.0,...),),...)

INIDEP=CREA_CHAMP(TYPE_CHAM='NOEU_DEPL_R',
                  OPERATION='ASSE',
                  MODELE=MOD[i],
```

```
                   OPTION='DEPL',
                   PROL_ZERO='OUI',
                   ASSE=(_F(GROUP_MA='ELEM',
                            CHAM_GD=DEPBID,
                            CUMUL='OUI',
                            COEF_R=0.0),
                         _F(GROUP_MA='ELEM0',
                            CHAM_GD=DEP[i-1],
                            CUMUL='OUI',
                            COEF_R=1.0,),),);

 SIEBID=CREA_CHAMP(TYPE_CHAM='ELGA_SIEF_R',
                   OPERATION='AFFE',
                   MODELE=MOD[i],
                   AFFE=_F(GROUP_MA='ELEM',
                            NOM_CMP=('SIXX','SIYY','SIZZ',...),
                            VALE=(1.0,1.0,1.0,...),),...)

 INISIG=CREA_CHAMP(TYPE_CHAM='ELGA_SIEF_R',
                   OPERATION='ASSE',
                   MODELE=MOD[i],
                   PROL_ZERO='OUI',
                   ASSE=(_F(GROUP_MA=ELEM,
                             CHAM_GD=SIEBID,
                             CUMUL='OUI',
                             COEF_R=0.0),
                          _F(GROUP_MA='ELEM0',
                             CHAM_GD=SIG[i-1],
                             CUMUL='OUI',
                             COEF_R=1.0,),),);

CART = CREA_CHAMP(TYPE_CHAM='ELGA_NEUT_R',
                  OPERATION='AFFE',
                  MODELE=MOD[i],
                  PROL_ZERO='OUI',
                  AFFE=_F(GROUP_MA=ELEM,
                          NOM_CMP =('X1','X2','X3', ...),
                          VALE    =(0.0,0.0,0.0,...),),)

INIVAR=CREA_CHAMP(TYPE_CHAM='ELGA_VARI_R',
                  OPERATION='ASSE',
                  MODELE=MOD[i],
                  PROL_ZERO='OUI',
                  ASSE=(_F(GROUP_MA=ELEM,
                           CHAM_GD=CART,
                           NOM_CMP =('X1','X2','X3',...),
                           NOM_CMP_RESU=('V1','V2','V3',...),
                           CUMUL='OUI', COEF_R=0.0),
                        _F(GROUP_MA=ELEM0,
                           CHAM_GD=VAR[i-1],
                           CUMUL='OUI', COEF_R=1.0),))

U[i]=STAT_NON_LINE(MODELE=MOD[i], CHAM_MATER=MAT[i],
                   ETAT_INIT=_F(SIGM=INISIG,VARI=INIVAR,DEPL=INIDEP),
                   EXCIT=_F(CHARGE=CHA[i]),...)

DEP[i]=CREA_CHAMP(TYPE_CHAM='NOEU_DEPL_R',
                  OPERATION='EXTR',
                  RESULTAT=U[i],
                  NOM_CHAM='DEPL',);
```

```
SIG[i]=CREA_CHAMP(TYPE_CHAM='ELGA_SIEF_R',
                  OPERATION='EXTR',
                  RESULTAT=U[i],
                  NOM_CHAM='SIEF_ELGA',);

VAR[i]=CREA_CHAMP(TYPE_CHAM='ELGA_VARI_R',
                  OPERATION='EXTR',
                  RESULTAT=U[i],
                  NOM_CHAM='VARI_ELGA',);
```

- **Suppression des concepts `INIVAR,INIDEP,INISIG,CART`**

```
DETRUIRE(CONCEPT=(_F(NOM=INIDEP),
                  _F(NOM=INISIG),
                  _F(NOM=INIVAR),
                  _F(NOM=CART),),)
```

2 Comment définir un état de contraintes initiales (ou de variables internes) avec des formules analytiques

2.1 Objectif

L'objectif est de pouvoir fabriquer deux des champs constituant l'état initial d'un calcul non_linéaire : le champ de contraintes et le champ de variables internes.
Dans les deux cas, la solution consiste à enchaîner un certain nombre de commandes `CREA_CHAMP`.

Le lecteur est invité à consulter la documentation `U2.01.09` (« Définition analytique d'un champ de contraintes et d'un champ de variables internes initiaux ») qui répond explicitement à cette problématique. Il peut regarder également le fichier de commande du cas-test `zzzz130a` qui illustre cette mise en œuvre.

Nous présentons ci-dessous une recette « succincte » pour chacun des cas.

2.2 Fabrication d'un champ de contraintes analytique

2.2.1 Recette

- Définir les formules : à partir de l'expression analytique de chaque contrainte, on construit le concept Aster correspondant à partir de l'opérateur `FORMULE`.
  Exemple :
  ```
  SIZZ = FORMULE (REEL='''(REEL:Z) = RHO*G*Z''').
  ```
  (la formule ainsi définie dépend de la variable `Z` (coordonnée géométrique)).

- Création du champ de formules : on construit un champ neutre (sans grandeur particulière) pour associer à chaque formule précédemment définie, une composante de ce champ.
  Exemple :
  ```
  SIZF=CREA_CHAMP(OPERATION='AFFE',TYPE_CHAM='ELGA_NEUT_F',
  AFFE=_F(TOUT=(OUI','NOM_CMP'='X1',VALE_F=SIZZ),...);
  ```

- Création du champ de paramètres :
  Nous devons disposer d'un champ dont les composantes font parties des variables des formules du champ d'évaluation. Dans cet exemple, nous devons disposer du champ géométrique (composante Z) aux points de Gauss. On procède en deux étapes :
  - extraction du champ géométrique aux nœuds :
    ```
    CHXN = CREA_CHAMP(OPERATION='EXTR',
                      TYPE_CHAM='NOEU_GEOM_R',NOM_CHAM='GEOMETRIE',...)
    ```
  - transformation du champ aux nœuds en un champ géométrique aux points de Gauss:
    ```
    CHXG = CREA_CHAMP(OPERATION='DISC',
                      TYPE_CHAM='ELGA_GEOM_R',...)
    ```

- Création du champ par évaluation des fonctions :
  A présent, on peut évaluer le champ : on utilise l'opération `EVAL` :
  ```
  SIGZ2=CREA_CHAMP(OPERATION='EVAL', TYPE_CHAM='ELGA_NEUT_R',
                   CHAM_F=SIZF, CHAM_PARA=CHXG, …)
  ```

- Création du champ de contraintes : pour obtenir notre champ de contraintes analytique, il faut attribuer la grandeur `SIEF_R` au champ d'évaluation.
  ```
  SIGZ=CREA_CHAM(OPERATION='ASSE', TYPE_CHAM='ELGA_SIEF_R',
                 ASSE=_F(CHAM_GD=SIGZ2,
                         NOM_CMP='X1',NOM_CMP_RESU='SIZZ'),...).
  ```

2.2.2 Commandes utilisées

Commandes	Fonctionnalité
`FORMULE`	Création des formules
`CREA_CHAMP/OPERATION='AFFE'`	Création du champ de formules
`CREA_CHAMP/OPERATION='EXTR'`	Création du champ de paramètres
`CREA_CHAMP/OPERATION='EVAL'`	Création du champ d'évaluation
`CREA_CHAMP/OPERATION='DISC'`	Création du champ de contraintes désiré après changement de grandeur

Tableau 2.1 :

2.3 Fabrication d'un champ de variables internes non nul

Deux méthodes sont envisageables pour créer un champ de variables internes non nul.

2.3.1 Première méthode

2.3.1.1 Recette

Cette méthode nécessite la connaissance du nombre de variables internes de chaque loi de comportement de votre STAT_NON_LINE. La doc U4.51.11 permet d'obtenir cette information. Cette méthode repose sur l'affectation de toutes les variables internes des lois fournies à votre STAT_NON_LINE. On utilise l'opérateur d'affectation AFFE de CREA_CHAMP pour cet effet.
Cette opération s'effectue en deux étapes:

- on crée d'abord une CARTE de NEUT_R
  Exemple :
```
VAIN = CREA_CHAMP(OPERATION='AFFE',,TYPE_CHAM='CART_NEUT_R',
                  AFFE=(_F(GROUP_MA ='GM1',
                           NOM_CMP=('X1','X2'),VALE=(0,,2,)),
                        _F(GROUP_MA ='GM2',
                           NOM_CMP=('X1','X2','X3',X4','X5'),
                           VALE=(0,0,0,9,2,1)),...)
```
- On transforme la carte de NEUT_R en champ aux point de Gauss de VARI_R via l'opération 'ASSE':
```
VARI=CREA_CHAMP(OPERATION='ASSE', TYPE_CHAM='ELGA_VARI_R',
                ASSE=(_F(CHAM_GD=VAIN,
                         GROUP_MA='GM1',NOM_CMP=('X1','X2'),
                         NOM_CMP_RESU=('V1','V2'))
                      _F(CHAM_GD=VAIN,
                         GROUP_MA='GM2',
                         NOM_CMP=('X1','X2','X3','X4','X5'),
                         NOM_CMP_RESU=('V1','V2','V3','V4','V5'),),),
                ...)
```

Le champ de variables internes VARI contient 2 variables internes pour les éléments du groupe $GM1$ et 5 pour $GM2$. Ainsi la loi de comportement à appliquer sur le groupe $GM1$ du modèle dans l'opérateur de résolution devra comporter exactement 2 variables internes.

2.3.1.2 Commandes utilisées

Commandes	Fonctionnalité
CREA_CHAMP/AFFE	Création d'une carte où toutes les composantes sont affectées.
CREA_CHAMP/OPERATION='ASSE'	Transformation de la carte en champ de variables internes.

2.3.2 Deuxième méthode

2.3.2.1 Recette

Cette méthode permet de n'affecter explicitement que les mailles qui possèdent des composantes non nulles. On procède de la façon suivante:

- On réalise un premier calcul bidon (pour bénéficier à postériori d'un champ de variables internes « modèle »):
```
UBID=STAT_NON_LINE(COMP_INCR=(_F(GROUP_MA='GM1',RELATION=...),
                              _F(GROUP_MA='GM2',RELATION=...),), …)
```
- On extrait le champ de variables internes VBID du résultat UBID:
```
VBID=CREA_CHAMP(OPERATION='EXTR',RESULTAT=UBID,
                TYPE_CHAM='ELGA_VARI_R',NOM_CHAM='VARI_ELGA',...)
```
- On affecte cette fois les valeurs non-nulles :
```
VAIN1 = CREA_CHAMP(OPERATION='AFFE',,TYPE_CHAM='CART_NEUT_R',
                   AFFE=(_F(GROUP_MA='GM1',
                            NOM_CMP=('X2'),VALE=(2,)),
                         _F(GROUP_MA ='GM2',
                            NOM_CMP=('X3',X4','X5'),VALE=(9,2,1)),),
                   ...)
```
- On met à zéro le champ de variables internes « modèle » et on le surcharge des valeurs non nulles:
```
VAIN2 = CREA_CHAMP(OPERATION='ASSE',TYPE_CHAM='ELGA_VARI_R'
                   ASSE=(_F(TOUT='OUI',CHAM_GD=VBID,
                            CUMUL='OUI',COEF_R=0.),
                         _F(GROUP_MA='GM1',CHAM_GD=VAIN1,
                            CUMUL='OUI',COEF_R=1.,
                            NOM_CMP=('X2'),NOM_CMP_RESU=('V2'),),
                         _F(GROUP_MA='GM2',CHAM_GD=VAIN1,
                            CUMUL='OUI',COEF_R=1.,
                            NOM_CMP=('X3','X4','X5'),
                            NOM_CMP_RESU=('V3','V4','V5'),),),...)
```

2.3.2.2 Commandes utilisées

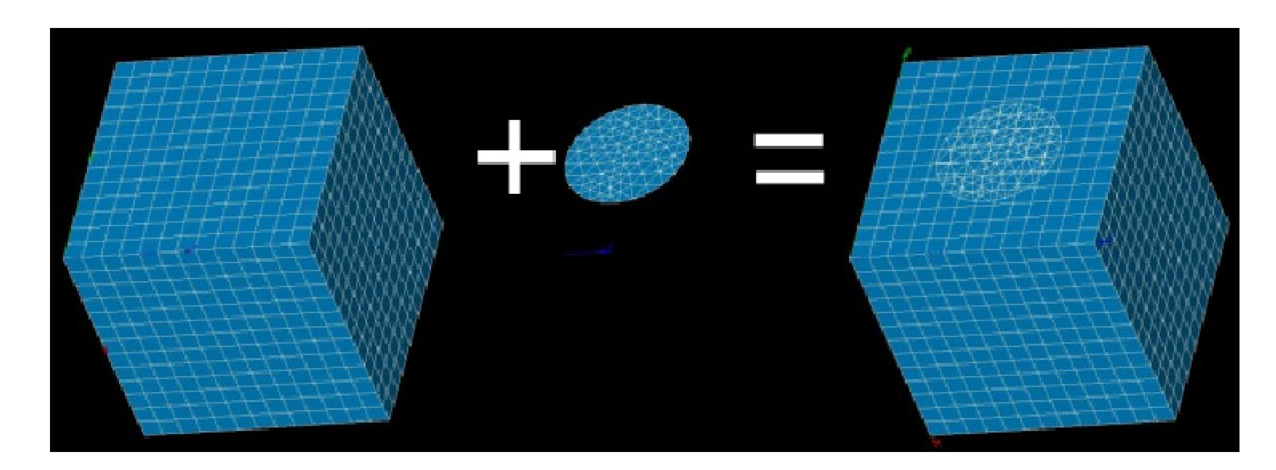
Commandes	Fonctionnalité
STAT_NON_LINE	Création d'un résultat bidon
CREA_CHAMP/EXTR	Extraction du champ de variables internes.
CREA_CHAMP/AFFE	Création d'une carte où seules les composantes non-nulles sont affectées.
CREA_CHAMP/ASSE	Transformation de la carte en champ de variables internes afin de l'assembler avec le champ extrait du

	résultat (préalablement remis à zéro).

3 Comment appliquer une pression sur une partie restreinte (non maillée exactement) du bord d'une structure 3D

Il est possible avec *Code_Aster* d'affecter un champ de pression sur une zone géométrique de la pièce à calculer non nécessairement maillée. En effet, il suffit de créer le maillage de peau support de la zone de pression, d'y créer un champ de pression qu'on projette sur le maillage de la pièce. La mise en donnée est alors semblable à la récupération / projection d'un champ de pression calculé par un code de CFD.

Dans l'exemple ci-dessous, on veut affecter une pression sur une zone circulaire centrée sur une face d'un cube, uniformément maillé en `HEXA8`. La maillage du disque est réalisé à part.



Remarque :

On vérifie la valeur de la résultante totale d'effort par un calcul de l'intégrale de pression sur les zones affectées. En effet, l'opération de projection induit une perte (proportionnelle à la grossièreté du maillage) qu'il convient de mesurer.

Nous allons nous intéresser à la création de la charge `CHA_PROJ`. Les différentes étapes sont les suivantes:

- lecture du maillage (disque) porteur de l'effort de pression
- création du champ de pression sur le disque
- projection du champ de pression depuis le disque vers la structure calculée
- création de la charge

Nous allons détailler chacune de ces étapes.

3.1 Étape 1 : Création du champ de pression sur le disque

- **Création d'un champ de pression constant par élément**

```
VAL =CREA_CHAMP( TYPE_CHAM =   'ELEM_PRES_R',
                 OPERATION =   'AFFE',
                 MODELE    =   modele2,
                 PROL_ZERO =   'OUI',
                 AFFE   =_F(   GROUP_MA = 'disque',
                               NOM_CMP  = 'PRES',
                               VALE     =  100000000.   ),
                 ...)
```

- **Création d'un résultat de type `EVOL_CHAR` à partir de ce champ**

```
RES_PRES=CREA_RESU(OPERATION = 'AFFE',
                   TYPE_RESU = 'EVOL_CHAR',
                   NOM_CHAM  = 'PRES',
                   AFFE      = _F(CHAM_GD   = VAL ,
                                  MODELE    = modele2,
                                  INST      = 0.     ), )
```

3.2 Étape 2 : Projection du champ de pression depuis le disque vers la structure calculée

```
RES_PROJ=PROJ_CHAMP(METHODE='ELEM',
                    RESULTAT=RES_PRES,
                    MODELE_1=modele2,
                    MODELE_2=MODE,
                    DISTANCE_MAX=1.,
                    CAS_FIGURE='2.5D',
                    PROL_ZERO='OUI',
                    VIS_A_VIS=_F(GROUP_MA_1='disque',
                                 GROUP_MA_2='Group_1',),
                    TOUT_ORDRE='OUI',);
```

3.3 Création de la charge

Pour la création du chargement, on utilise les chargements évolutifs dans le temps de type '`evol_char`' produits par `LIRE_RESU` [U7.02.01] (ou `CREA_RESU`), et contenant des champs de pression, des densités de force volumique en 2D ou 3D et des densités de force surfacique en 2D ou 3D.

```
CHA_PROJ=AFFE_CHAR_MECA(MODELE=MODE,
                        EVOL_CHAR=RES_PROJ,)
```

3.4 Fichier de commandes

```
DEBUT()

MAIL=LIRE_MAILLAGE(FORMAT='MED',);

MAIL=MODI_MAILLAGE(reuse =MAIL,
                    MAILLAGE=MAIL,
                    ORIE_PEAU_3D=_F(GROUP_MA='Group_1',),)

MAIL=DEFI_GROUP(reuse =MAIL,
                MAILLAGE=MAIL,
                CREA_GROUP_NO=_F(NOM='Group_3',
                                 GROUP_MA='Group_3' ) )

MA=DEFI_MATERIAU(ELAS=_F(  E = 2.E11,
                           NU = 0.3,
                           ALPHA = 1.E-5,
                           RHO=1.  ),
                 ECRO_LINE=_F(  D_SIGM_EPSI = 2.E9,
                                SY = 2.E8 )   )

MODE=AFFE_MODELE(MAILLAGE=MAIL,
                 AFFE=_F(TOUT='OUI',
                         PHENOMENE='MECANIQUE',
                         MODELISATION='3D',),);

MATE=AFFE_MATERIAU(MAILLAGE=MAIL,
                   AFFE=_F(TOUT='OUI',
                           MATER=MA,),);

##########################################
##### création de la charge CHA_PROJ #####
##########################################

# lecture du maillage (disque) porteur de l effort de pression
mail2=LIRE_MAILLAGE(UNITE=21,
                    FORMAT='MED',);

mail2=MODI_MAILLAGE(reuse =mail2,
                    MAILLAGE=mail2,
                    ORIE_PEAU_3D=_F(GROUP_MA='disque',),);

modele2=AFFE_MODELE(MAILLAGE=mail2,
                    AFFE=_F(TOUT='OUI',
                            PHENOMENE='MECANIQUE',
                            MODELISATION='3D',),)

# création du champ de pression sur le disque
VAL =CREA_CHAMP(TYPE_CHAM='ELEM_PRES_R',
          OPERATION='AFFE',
          MODELE=modele2,
          PROL_ZERO='OUI',
          AFFE=_F(GROUP_MA = 'disque',
             NOM_CMP  = 'PRES',
             VALE     =  100000000.   ),
          INFO=1,)

RES_PRES=CREA_RESU(OPERATION = 'AFFE',
```

```
                    TYPE_RESU = 'EVOL_CHAR',
                    NOM_CHAM  = 'PRES',
                    AFFE      = _F(CHAM_GD   = VAL ,
                                  MODELE    = modele2,
                                  INST      = 0.      ), )

# calcul de la force de pression résultante sur le disque du premier
maillage
tab1=POST_ELEM(INTEGRALE = _F(GROUP_MA = 'disque',
                              DEJA_INTEGRE = 'NON',
               NOM_CHAM = 'PRES',
               NOM_CMP  = 'PRES'  ),
        RESULTAT  = RES_PRES ,
        MODELE    = modele2   )

IMPR_TABLE(TABLE=tab1)

# projection du champ de pression depuis le disque vers la structure
calculee
RES_PROJ=PROJ_CHAMP(METHODE='ELEM',
                    RESULTAT=RES_PRES,
                    MODELE_1=modele2,
                    MODELE_2=MODE,
                    DISTANCE_MAX=1.,
                    CAS_FIGURE='2.5D',
                    PROL_ZERO='OUI',
                    VIS_A_VIS=_F(GROUP_MA_1='disque',
                                 GROUP_MA_2='Group_1',),
                    TOUT_ORDRE='OUI',);

# calcul de la force de pression résultante après projection
tab2=POST_ELEM(INTEGRALE = _F(GROUP_MA = 'Group_1',
               NOM_CHAM = 'PRES',
                              DEJA_INTEGRE = 'NON',
               NOM_CMP  = 'PRES'  ),
        RESULTAT  = RES_PROJ ,
        MODELE    = MODE   )

IMPR_TABLE(TABLE=tab2)

# création de la charge
CHA_PROJ=AFFE_CHAR_MECA(MODELE=MODE,
                        EVOL_CHAR=RES_PROJ,)

##################################################
##### fin de creation de la charge CHA_PROJ #####
##################################################

BLOQ=AFFE_CHAR_MECA(MODELE=MODE,
                    DDL_IMPO=_F(GROUP_MA='Group_3',
                                DX=0.0,
                                DY=0.0,
                                DZ=0.0,),)

RESU=MECA_STATIQUE(MODELE     = MODE,
          CHAM_MATER = MATE,
          EXCIT=(_F(CHARGE=BLOQ,),
            _F(CHARGE=CHA_PROJ,),) )

RESU=CALC_NO(reuse =RESU,
```

```
                OPTION='REAC_NODA',
                RESULTAT=RESU,)

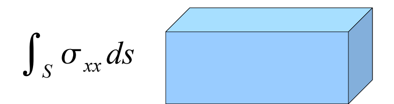
STANLEY()

FIN()
```

4 Comment calculer l'intégrale des contraintes sur le bord d'une structure 3D

4.1 Objectif

L'utilisateur dispose d'un champ de contraintes (aux points de Gauss). Il souhaite faire l'intégrale de chaque composante sur le bord de son modèle.



Le calcul de l'intégrale est fait par la commande `POST_ELEM`. Le problème est que ce calcul nécessite actuellement que la quantifié à intégrer soit discrétisée sur les points de Gauss des éléments qui calculent l'intégrale (ici les éléments de bord de type facette).

Les contraintes aux points de Gauss ne sont pas calculées sur les faces, aussi l'utilisateur se voit confronté à quelques difficultés. Une recette a été établie pour pouvoir réaliser ce calcul. Elle est présentée ci-dessous.

4.2 Recette

4.2.1 Étape 1 : Calcul du champ de contraintes 3D aux nœuds

Tout d'abord, il faut calculer les champs de contraintes aux nœuds. Pour cela, on calcule au préalable les champs de contraintes aux nœuds par éléments.

1) calcul du champ de contraintes aux nœuds par élément :
```
CALC_ELEM(OPTION='SIEF_ELNO_ELGA',RESULTAT=...,)
```

2) calcul du champ de contraintes aux nœuds :
```
CALC_NO(OPTION='SIEF_NOEU_ELGA',RESULTAT=...,)
```

L'étape suivante consiste à construire un champ neutre dont les valeurs :
- sont localisées aux points de Gauss des mailles (2D et 3D), et
- correspondent aux valeurs des contraintes.

4.2.2 Étape 2 : Calcul d'un champ neutre 'ELGA' correspondant aux contraintes

1) On suppose que le modèle 3D (`MO3D`) qui a servi à faire le calcul contient en plus des éléments « 3D », des éléments de bord (facettes) sur tous ses bords.

2) Extraction du champ de contraintes aux nœuds:
```
SIGNO=CREA_CHAMP(OPERATION='EXTR', TYPE_CHAM='NOEU_SIEF_R',
                 NOM_CHAM='SIEF_NOEU_ELGA',RESULTAT=...,INST=...)
```

3) Transformation du champ `SIGNO` en champ neutre:
```
NEUTNO=CREA_CHAMP(OPERATION='ASSE', TYPE_CHAM='NOEU_NEUT_R',MODELE=MO3D,
                  PROL_ZERO='OUI',ASSE=_F(TOUT='OUI',CHAM_GD=SIGNO,
                  NOM_CMP=('SIXX','SIYY','SIZZ'),
                  NOM_CMP_RESU=('X1','X2','X3'),)
```

4) Passage du champ `NEUTNO` aux points de Gauss:
```
NEUTGA=CREA_CHAMP(OPERATION='DISC',TYPE_CHAM='ELGA_NEUT_R',
                  CHAM_GD=NEUTNO, MODELE=MOD3D, PROL_ZERO='OUI')
```

4.2.3 Étape 3 : Calcul de l'intégrale sur le bord

Il suffit d'intégrer le champ NEUTGA aux mailles de bord. Par exemple, pour la face 'HAUT', la commande est la suivante:

```
INT=POST_ELEM(CHAM_GD=SIGFA, MODELE=MOD3D,
              INTEGRALE=_F(GROUP_MA='HAUT',
                           NOM_CMP=('X1','X2','X3'),),)
```

4.3 Commandes utilisées

Commandes	Fonctionnalité
CALC_ELEM	Création des champs de contraintes aux nœuds par éléments aux différents instants.
CALC_NO	Création des champs de contraintes aux nœuds aux différents instants.
AFFE_MODELE	Création d'un modèle 3D.
CREA_CHAMP/EXTR	Extraction du champ de contraintes aux nœuds à l'instant INST.
CREA_CHAMP/ASSE	Transformation du champ de contraintes en champ neutre.
CREA_CHAMP/DISC	Passage du champ neutre aux nœuds au champ neutre aux points de Gauss.
POST_ELEM	Calcul de l'intégrale en spécifiant les groupes de mailles de face.